PML Assignment

scott

Thursday, November 06, 2014

```
setwd("C:\\Users\\srobin\\Documents\\GitHub\\machineLearning")

library(knitr)
```

Weight Lifting Exercises Dataset

This dataset was derived from weight lifting exercises of six healthy subjects, equiped with sensing devices on-body and within the dumbbell used to do the exercise.

“Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E).”

Class A was considered to be the correct manner in doing the exercise, while Class B,C,D and E were characteristic of incorrect procedures.

Read more: http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz3IbSKQZIv (http://groupware.les.inf.puc-rio.br/har#weight_lifting_exercises#ixzz3IbSKQZIv)

The goal of this analysis is to predict the “classe” of 20 test cases solely from the manner in which they did the exercise, represented by the data within those 20 cases.

```
# read in data and convert missinga info to "NA"
train <- read.csv("pml-training.csv",na.strings=c("", "NA"))
test <- read.csv("pml-testing.csv",na.strings=c("", "NA"))

# throw out all cols with NA or missing values
trainDel <- train[,colSums(is.na(train)) == 0 ]
testDel <- test[,colSums(is.na(test)) == 0 ]
```

Preparation

Parrallel processing was set up using 7 of the 8 CPU cores on the test system - Lenovo W530 laptop w/8GB memory and an i7-3630QM CPU at 2.40GB.

Loading data and exploration

The data was read using the -na.strings- parameter and then a subset was created to omit all columns with NA. This trimmed the number of columns (variables) from 160 to 60. A summary of the data was reviewed to ensure all observations were valid and did not include NAs. There were 19622 observations. Data was split with -createDataPartition- into a 60/40 split (Training vs. Testing).

A basic “gbm” model (defaults) was run using caret -train- on the training dataset variables using “classe” as the class variable, then -varImp()- was applied to the model to determine the most important variables. The top 22 variables were selected to be used in the final model. 22 is an arbitrary selection, as the major issue was to reduce the number of variables to scale down the model for quicker processing, as well as to denote the variables that have the most influenence on the predictions.

```
# Set up Parrallel processing to improve speed library(doSNOW) cl <-
# makeCluster(7, type='SOCK') registerDoSNOW(cl) split for training and
# testing library(caret) set.seed(1235) inTrain <-
# createDataPartition(y=trainDel$classe, p=0.60,list=FALSE) training <-
# trainDel[inTrain,]; testing <- trainDel[-inTrain,] model modelFit <-
# train(classe~., data=training, method='gbm' ) stop parrallel processing
# and reclaim memory stopCluster(cl)

# varImp(modelFit)
```

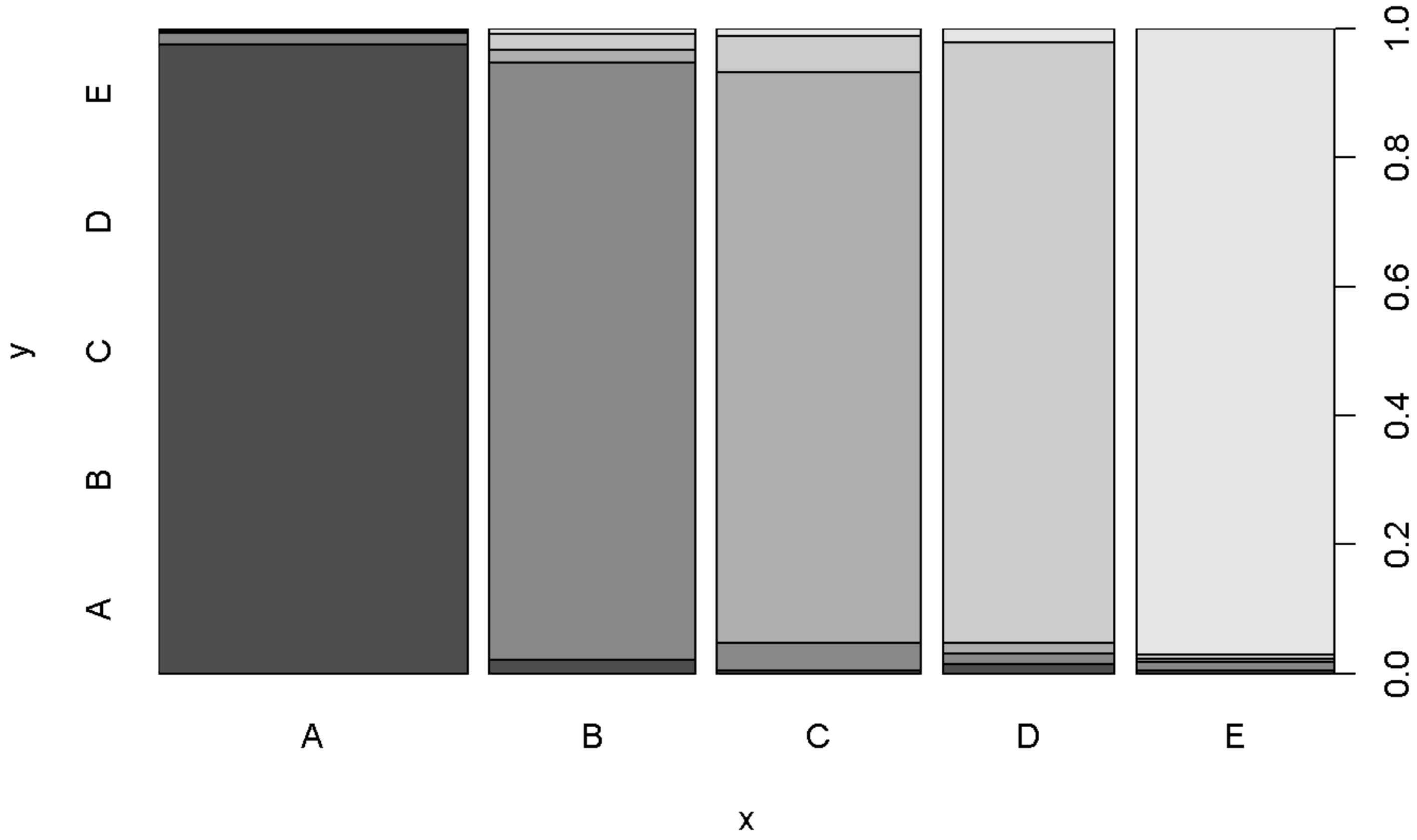
Using the model

K-fold cross validation was selected. 10 folds, repeated 10 times, with threshold of 0.80 was used. Preprocessing using “center”, “scale” and “pca”.

Variables were reduced to 20 variable for the final model.

```r
library(caret)
# selected variables from varImp()

myVars <- c( "num_window","yaw_belt","pitch_belt","accel_forearm_x","roll_belt",
             "roll_forearm","magnet_dumbbell_z","pitch_forearm",
             "magnet_dumbbell_x","magnet_dumbbell_y","magnet_belt_x",
             "accel_dumbbell_y","roll_arm","roll_dumbbell","magnet_arm_y",
             "magnet_belt_z","accel_arm_x","magnet_forearm_y",
             "accel_belt_z","magnet_forearm_x","accel_dumbbell_z",
             "classe"
             )


set.seed(1)  # produced 100% correct
inTrain <- createDataPartition(y=trainDel$classe,
                               p=0.60,list=FALSE)

trainingRF <- trainDel[inTrain,myVars]
testingRF  <- trainDel[-inTrain,myVars]

# Set up parrallel processing
library(parallel)
cl2 <- makeCluster(7, type="SOCK")
registerDoSNOW(cl2)

# trainControl
folds <- 10
repeats <- 10
fitControl <- trainControl(method="repeatedcv",
                           number=folds,
                           repeats=repeats,
                           #classProbs=TRUE,
                           preProcOptions=list(thresh=0.8),
                           allowParallel=TRUE
                          )

# model - preprocessing and tuning
Fit_rf <- train(classe ~ .,
                     data=trainingRF,
                     trControl=fitControl,
                     tuneLength=5,
                     preProcess=c("center","scale","pca"),
                     method="rf"
)

# Shut off parrallel processing and reclaim memory
stopCluster(cl2)
```

```
predictions_rf <- predict(Fit_rf,testingRF)

plot(predictions_rf,testingRF$classe)
```



Analysis the results

```
# true accuracy of the predicted model
outOfSampleError <- sum(predictions_rf == testingRF$classe)/length(predictions_rf)
outOfSampleError
```

```
## [1] 0.9414
```

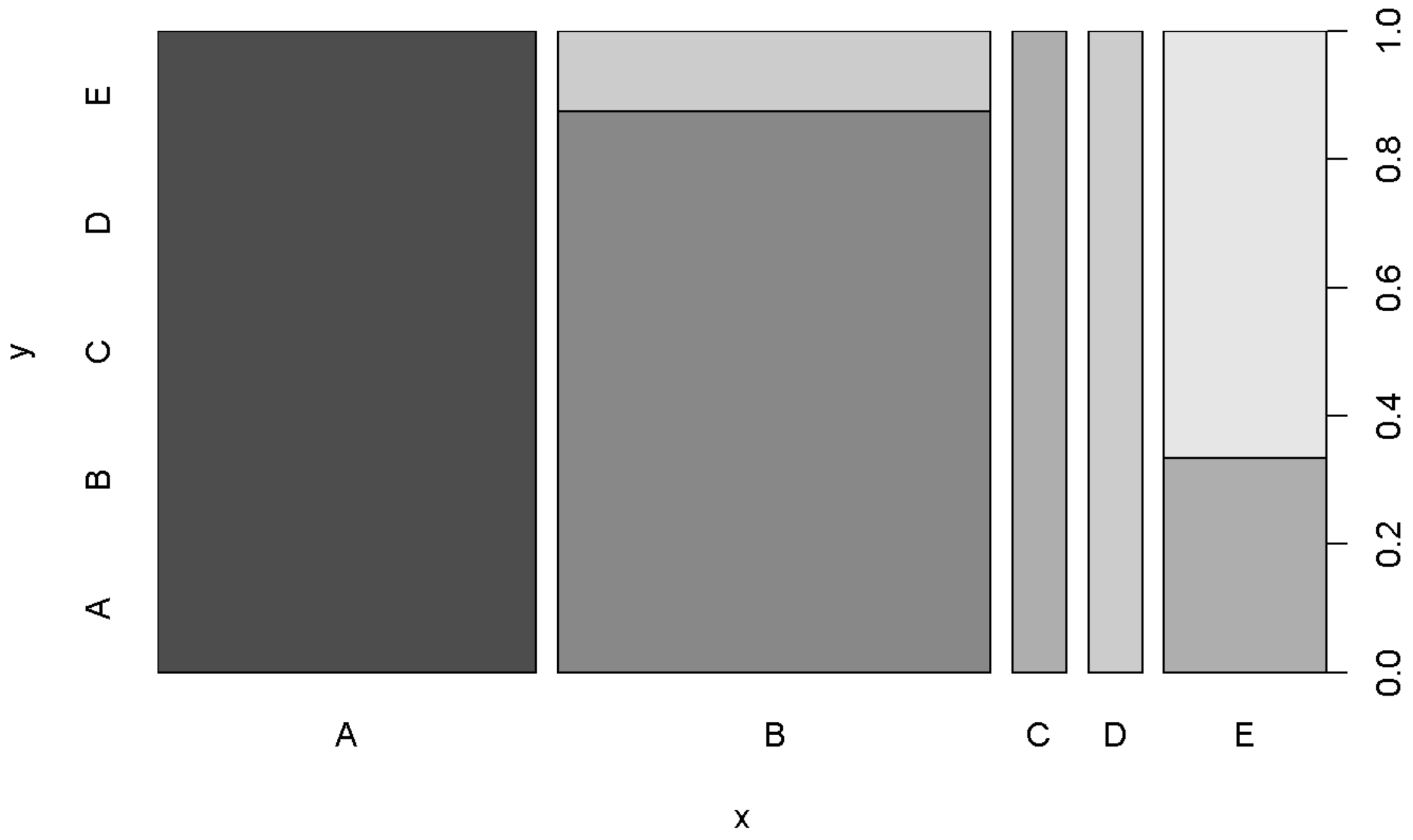
```
# out of sample error and percentage of out of sample error
outOfSampleError <- 1 - outOfSampleError
outOfSampleError
```

```
## [1] 0.05863
```

```
correct <- as.factor(c("B","A","B","A","A","E","D","B","A","A","B","C","B","A","E","E","A","
B","B","B"))
pred <- predict(Fit_rf,test)
pred
```

```
##  [1] B A B A A C D B A A B C B A E E A D B B
## Levels: A B C D E
```

```
plot(correct,pred)
```


y
A
B
C
D
E
1.0
0.8
0.6
0.4
0.2
0.0
A
B
C
D
E
x